MYDOS Version 4 User Guide
Revision 4.50
for Atari Home Computers

Charles Marslett
2705 Pinewood Dr.
Garland, TX 75042
CIS: 73317,3662
UseNet: CHASM@KILLER.DALLAS.TX.US

and

Robert Puff
Suite 222
2117 Buffalo Rd.
Rochester, NY 14624
GEnie: BOB.PUFF

I. INTRODUCTION

The disk operating system described in this manual is modeled after the ATARI (trademark of ATARI Corp.) disk operating system (DOS 2), and may be considered an extension of the very "user friendly" concepts introduced with this operating system. The ATARI 810 disk drive and the ATARI 1050 disk drive are well supported by the DOS 2 and DOS 2.5, but those DOSs have very limited provision for higher capacity double density disks (such as the XF-551), and large RAMdisks.

II. SYSTEM REQUIREMENTS

MYDOS 4.50 is intended to function as nearly as possible like ATARI DOS 2. This means it occupies as little memory as possible, supports all the system calls supported by DOS 2, and in most cases uses exactly the same parameters to the system calls. This means compatibility, yet MYDOS goes far beyond this. On a standard single-sided, single density disk, MYDOS will read and write to a DOS 2.0 or 2.5 format, and vice versa. There are a few programs that will not run under MYDOS because of one or both of the following reasons: direct "illegal" calls into the main DOS or FMS code, and relying upon exact directory data (i.e., 3 digit sector counts).

MYDOS will run on any 8-bit Atari computer with at least one disk drive (that is compatible with DOS 2). MYDOS 4.50 has full READ compatibility with the DOS 2.5 format, but will only write to the first 720 sectors of the disk. (MYDOS has its own format for enhanced density disks that allow full access.) It is not compatible at this time with the OS/A+ version 4 format, or the SpartaDOS format.

Up to eight disk drives (or nine if the ninth is a RAMdisk) may be accessed, but only one is required. The resident part of the operating system (DOS.SYS) supports all documented functions of the DOS 2 operating system, so MYDOS supports most available software for the ATARI home computers.

The memory available to a program is affected only by the number of files to be open concurrently: each disk file that may be open at the same time requires 256 bytes of buffer space. Unlike Atari DOS 2, memory requirements are independent of the number of disk drives or the sector size and density. The need for a 256 byte buffer for each concurrently open file means that programs that keep large numbers of files open will have less memory available than under DOS 2 and as a result some programs (mostly older versions of compilers and language interpreters) will not have enough memory to run. With three file buffers (which is the default) and no resident drivers (such as for RS232 support), MYDOS 4.50 permits binary programs to load as low as $1F00.

MYDOS 4.50 also contains a very versatile RAMdisk driver than can handle upgrades up to 1 megabyte that follow the guidelines of the expanded memory on the 130XE, or the Axlon upgrade for the 800. MYDOS also has a feature that will determine how much memory you have, and set itself up for your machine (For XE-compatible upgrades only) automatically.

III. MENU FUNCTIONS

The menu provided by MYDOS 4.50 identifies 18 common tasks that might need to be done. Rather than having to write a utility program (only a few lines of BASIC would perform most of the menu functions) or even remember the name and format of a DOS command, these tasks can be handled by entering a single letter. MYDOS responds with a question asking for the details of the operation (which file, what density, "are you sure?" or whatever else it might need to know). After you enter the remaining information, the function is performed and another prompt is displayed.

You should notice two interesting things about the menu: the second line on the screen identifies the disk drives present on the system and what they appear to MYDOS to be (single or double density, RAMdisk or high capacity hard disk). The next line describes the current default directory (that directory used when a disk is referenced by "D:" (without a unit number after the "D" - this feature will be described later). The second thing to notice is that after commands fill the screen, the menu "rolls" off the top: some DOS programs (e.g. Atari DOS 2) keep the menu, MYDOS does not. This permits more information to be displayed when a long sequence of commands are done, or when many files are copied. To restore the menu to the screen, just type RETURN, and the initial screen will be restored.

If you wish to abort a command at any point, simply press BREAK. This will return you to the prompt. Pressing BREAK or RESET while writing to a disk will not cause any damage; however, the file you were writing will probably not be displayed in the directory, and the free sector count may be incorrect.

When entering a filename, the drive number should be entered, followed by a ":", and then the rest of the filename. If the drive number and ":" are not entered, the default drive (and directory) will be used. In the [C]opy file option, entering the drive number only will result in copying every file on the main directory of that drive. A drive may be specified like this: "1", "1:", or "D1:". If you wish to specify the file or set of files to be referenced, the drive format must include a ":" or it must be omitted entirely (for the default drive and directory). Examples: D1:Test.obj, 1:TEST.ASM, or D2TEST (really D1:D2TEST) are valid file names, but d1:Test.obj or 1TEST.ASM are not.

The file name itself is either fully specified (referring to the

entire name as it is on the disk) or includes "wild card" characters (specifying a set of files). A fully specified file name consists of one to eight characters followed by a period (".") and zero to three additional "extender" characters. The characters in the file name may be upper or lower case letters, numbers, the underscore ("_") or the character "@". The only exception is the first character - it may not be a number. The "wild card" characters are the characters "*" and "?": the character "*" or the sequence ".*" end either the 8 character or the 3 character field in the file name and match all possible characters. The character "?" matches only one single file name character. Examples: "*.BAS" will match every file ending with .BAS. "TEST?.*" will match files TEST.BAS, TEST1.OBJ, but will not match TESTER.BAS.

In addition to the main directory (containing up to 64 files or directories), each MYDOS disk may also contain additional subdirectories of 64 files each. You may use one of two possible characters to separate the subdirectory name(s): the ":" colon, or the ">" greater-than sign. If the main directory contained the subdirectory BAS and the file GRAPHIC1 were in the subdirectory file BAS, it could be referenced with the filename BAS:GRAPHIC1 (or BAS>GRAPHIC1). If instead, GRAPHIC1 were in the subdirectory GR.dir which in turn were in BAS, then the reference would be to BAS:GR.dir:GRAPHIC1 (and so on with as many names as needed). Because there is no limit to the number of subdirectories on a disk (other than the buffer size of programs using the directories and number of available sectors on the disk), a single diskette can contain hundreds of files if necessary. Each subdirectory is a 8 sector file to its parent directory. Thus it is limited to 64 files or subdirectories like the 8 sector root directory.

If a disk directory includes the files TEST.ASM, TEST.OBJ, TEST.C, TEST.ALM, TEASET.DOC, TRACE.FIL, and BETS.LST, the specification "t*.*" will not match any file name (since "t" and "T" are not the same letter to MYDOS). The specification "T**" will match all but "BETS.LST" (since the others all begin with the letter "T"). The specification "?E??.*" will match the first four files and the last one (since the 8 character part of the file name must have no more than 4 characters in it and the second character must be an "E"). The specification "*.?" will match only the file TEST.C (since it is the only file name with a single character in the 3 character field). The specification "????E**" will match the files TEASET.DOC and TRACE.FIL and none of the others (since the 8 character part of the file name must have at least 5 characters and the fifth must be an "E"). Note that MYDOS does not always require the "." period to separate the extender from the 8 character filename. Examples: FILENAME.EXT could be entered as FILENAMEEXT; TES*.* could be TES**, but TEST.* could NOT be entered as TEST*.

IV. THE MENU COMMANDS

A. List a Directory or a Set of Files

The "A" command will list the files on a disk with their sizes, followed by a line specifying the number of free sectors on the disk. If the line starts with a "*", the file has been locked and may not be modified or deleted without first being unlocked. A ":" before the file name marks those files that are subdirectories. These files cannot be read or written as other files but only accessed as directories or deleted (if the directory is empty). File lengths and the number of free sectors are reported as 4 digit decimal numbers (most other Atari DOSs report the values as 3 digit numbers).

If the number of free sectors on a drive exceeds 9999, the free sector line will contain a 5 digit number, and be one character longer. If a file is longer than 9999 sectors, the blank between the file extender and its sector count will contain the most significant digit (the line size does NOT change).

No indication is made of the format of the file but ATARI DOS 2.0, 2.5, and MYDOS are the only three supported file formats. SpartaDOS or OS/A+ Version 4 files must be converted to single or double density Atari DOS 2 files first in order to be usable from MYDOS. See Section 6 for further directory information if you need more detail.

The "A" command will list the directory information to the screen if only one file specification is entered. If two are entered, the second is taken as a destination file and will be overwritten (or appended to if the /A flag is used) with the directory data. Use "2,D1:DIR.TXT" to save the directory data of drive 2 to a text file DIR.TXT on drive 1. The entry "1,P:" will give you a printout of the directory of drive 1.

To list the files in a subdirectory, enter the name of the directory followed by a ":" colon or ">" greater-than sign. For example, "1:TEST:BAS:" will list the files in the subdirectory BAS which in turn is in the subdirectory TEST in the main directory of the diskette in drive 1. Using "D1:TEST>BAS>" will achieve the same results.

B. Run the Cartridge

The "B" command returns control to the cartridge in the left (or only) cartridge slot. If no cartridge is present, an error is displayed, and nothing happens. No additional information is required, so if a cartridge is present, it is entered after loading MEM.SAV (if enabled) or immediately.

C. Copy a File or a Set of Files

The "C" command is used to make another copy of one or more files of data. The two file specifications asked for after entering the "C" identify the source and the destination of the information being copied. Either may be fully specified disk file or a device

specification (such as E:, P: or one of the RS232 ports R1: to R4:). The destination may be a set of disk files (specified with "*" and "?"s) only if the source specifies a file name for the destination to use. The source may be a set and the destination a single disk file, but unless the "/A" modifier is specified to append each copied file to the end of the previously copied files, only the last source file will remain on the destination disk.

Note that the "C" command always uses the full memory space for a copy operation (unlike ATARI DOS 2) and as a result, it will always invalidate MEM.SAV if it is used. Any pending program cannot be restarted after a "C" or "J" command.

There are several options when entering files to copy. As with all filename inputs, the default drive and directory (whatever "D:" is equal to on the top status line) will be used if Dn: is not specified. Using "/Q" at the end of the line will query you for each source file it finds (useful for copying certain files but specifying *.*) if you wish to copy it. Using "/S" will copy all files EXCEPT those with the first letter of the extender being "S" (will omit all *.S??). This emulates DOS 2.0's [O] copy file option, and is useful when you want to copy all files except .SYS files. The last option is the "/X" parameter. This will prompt you for disk swaps, even if the source and destination drives are not the same number. Here are some examples of using the [C]opy file(s) command:

"TEST.BAS" will copy the file TEST.BAS on the default drive (and directory) to the same drive, but prompt you for the destination. (Same as the [O] command in DOS 2.0.)
"D1:TE*.*,3" will copy all files on drive 1 starting with "TE" to drive 3.
"D4:DIR>*.TXT,E:/Q" will copy all files in the DIR directory of drive 4 ending with .TXT to the screen, but will query you before each one.
"2:ADD,2:LIST/A" will append the data in the file ADD on drive 2 to the file LIST on drive 2.

D. Delete a File or Set of Files

The "D" command will remove all files that match the file specification entered, asking for confirmation before each one is removed. The "Delete file xxx?" prompts may be disabled by adding "/N" to the filename. In this case, all the matching files will be removed "quietly", and the only further indication you will see is the prompt for the next command.

E. Rename a File or Set of Files

The "E" command changes the name of the source file or files to match the specification in the destination. Unlike other file specifications, the destination specification MUST consist of a single file name: it must not contain any directory names or a disk drive number. For example, "D2:TEST>BASIC>NOTPNT.BAS,RANDIO.BAS" is

the line entered to change the name of a file in the directories "D2:TEST>BASIC>". To change the name of the directory "BASIC" to "ATBASIC", the line would look like "D2:TEST>BASIC,ATBASIC". One word of caution: Never rename more than one file to another name without wildcards in the destination. For example, renaming "*.*" to "TEST" would rename EVERY file to "TEST", making it impossible to retrieve files other than the first one!

F. Lock a File or Set of Files

The "F" command limits access to the files identified. "Locked" files may not be deleted, renamed, added to, or replaced without being first unlocked with the "G" command. When a directory is listed, the files that have been locked will be marked with an "*" in the first column. The files that are locked may be read or loaded and executed normally, only modification or removal are prohibited. A locked directory cannot be deleted or renamed and no files may be written to it. The files already in it are not locked however.

MYDOS will not ask you for confirmation on each file, "Lock xxx?" unless you specify this feature by adding "/Q" to the filename.

G. Unlock a File or Set of Files

The "G" command removes the limitations imposed on a file when it is "locked" using the "F" command. It does not alter the file or otherwise change the way the file is accessed or used. The same function may be performed in a program through the CIO function to "unlock" a file.

As with the "F" function, MYDOS will not ask you for confirmation with each file to unlock, unless you specify the "/Q" parameter after the filename.

H. Write MYDOS to a Disk

The "H" command is used to make a bootable copy of the current MYDOS 4.50 files in memory. The two files created or rewritten are "DOS.SYS" and "DUP.SYS". "DOS.SYS" is the permanently resident file management routine accessed through CIO and the small interface package that loads and saves MEM.SAV (an image of the part of memory used to hold the nonresident part (DUP.SYS) of MYDOS), and the code to load the menu portion of MYDOS ("DUP.SYS"). The file "DUP.SYS" is a standard binary load file containing the part of MYDOS that is overwritten when a program is loaded into memory. Neither of these files may be interchanged with any other version or type of disk operating system; both should be treated as a single object. Never copy only DOS.SYS or only DUP.SYS to a disk without copying the other, unless you will never access the DUP.SYS menu (called by typing "DOS"). The files written to the disk by the "H" command will reflect the configuration parameters currently in memory, which may

be different from the ones active if the system were rebooted from the master disk again. (See Section IV.O., configuring the system, for the definition of the configuration parameters provided in the system and how to specify a modified configuration.)

I. Initialize a Diskette

The "I" command is used to "Format", or prepare a new disk for use with the MYDOS operating system, or to remove all the files on an old disk. The result of the "I" command is a completely empty disk. The only data on the diskette is that system provided information defining the space available and the empty main directory. If the drive number is followed by a "/N" parameter, the diskette will not be reformatted, but just "erased" (the directory sectors will be rewritten). If a disk has recently been formatted, this is the fastest way to remove all the files on a diskette, rather than to use the "D" command. Use this option ONLY if you know the disk was good, for this does not verify the integrity of the disk. Due to a bug in some hard disk interfaces, this "/N" parameter may be needed to initially format the hard disk partitions.

A diskette may also be formatted in the Atari 1050 enhanced format by responding to the question "(Type [A] to format in Enhanced) Type [Y] to format drive x:" with a "A" (presuming the drive is a 1050 drive or compatible with the 1050). The diskette is, however, not compatible with the diskettes similarly created by DOS 2.5. The MYDOS diskette will not be readable by DOS 2.5, but DOS 2.5 enhanced disks can be fully read by MYDOS.

An additional note: it is a good idea to set the density of the drive you are about to format, unless the drive is not capable of double density. (see the "P" command for information on setting density.) Otherwise, MYDOS will format the drive in the density of which it was last accessed.

J. Duplicate a Diskette

The "J" command copies all the information from one diskette to another. The information to be copied form the source diskette is determined by specifying a starting and an ending sector number. If the range is not stated, the sectors marked as in use in the DOS bit map (on the VTOC sectors) are copied. The VTOC may be constructed by Atari DOS 2, any version of MYDOS or any other compatible operating system.

Specifying a sector range is done by adding two numbers separated by a dash and enclosed in parentheses to the end of the drive specification(s). For example, to copy sectors 19 through 54 (tracks 1 through 3) from drive 1 to drive 3 without formatting the diskette in drive 3, the command line could be "1,3/N(19-54)". To copy the entire disk from drive 1 to drive 2 after formatting the diskette in drive 2 the command could be "1,2(1-720)". This is the

proper way to duplicate a non-MYDOS compatible diskette (of course, copies of "copy protected" disks will not normally be usable after copying but data disks for some word processors and games can be backed up this way). Enhanced density disks may be duplicated in the same manner: "1,2(1-1040)".

If the destination disk is already a properly formatted diskette, the "/N" modifier may be entered after either drive number to skip the formatting of the destination drive. Otherwise, the destination diskette will be formatted before the data from the source is copied to it. That is, either "1/N,2" or "1,2/N" will copy from drive 1 to drive 2 without first formatting the diskette in drive 2.

Note that the "J" command, like that in ATARI DOS 2, will use all of available memory to duplicate the diskette: this means that if memory has been saved using the MEM.SAV file, it will no longer be valid. Any pending program cannot be restarted after a "C" or "J" command.

K. Save Memory to Disk

The "K" command builds a binary load file containing the data from the memory area specified, as well as adding an initialization and a run vector address if specified. If the file is not to execute an initialization routine on being loaded, the initialization vector should be omitted. If it is not to run on being loaded, the run vector should also be omitted (trailing commas need not be entered). If either vector is entered as zero, that is equivalent to omitting it. Note that the starting and ending addresses of the program and both entry points are all specified as hex numbers. You may append data to another file by using the "/A" parameter. This command may also be used for just adding a Run address to a binary file. Example: "D1:PROGRAM.OBJ/A,0,0,,4000" would add the run address of $4000 to the program "PROGRAM.OBJ" on drive 1.

If MEM.SAV is active when the "K" command is entered, the MEM.SAV file is loaded before writing the file to the disk.

L. Load Memory from a File/Disable MEM.SAV

The "L" command takes a binary load file from the disk and loads it into memory. The load file's initialization routine(s) will be executed and the program started at its run address unless the "/N" parameter is added to the file name. If the file is not a binary file, you will be returned with an "Error -- 180". A binary file that is corrupted will return an "Error -- 181".

This command also disables the MEM.SAV feature before loading and executing the program. To just disable MEM.SAV, press "L", then [RETURN].

M. Run at Address

The "M" command is used to enter a program loaded without a run address, or to jump into any program without the need for a return address. It may be used to restart the computer (loading the AUTORUN.SYS file, if any, or a new DOS, etc...) by specifying "E477" as the jump address. If MEM.SAV is active (enabled with the "N" command and not since disabled by the "L" command), the contents of memory will be restored from MEM.SAV before jumping to the address specified.

N. Load MEM.SAV from a File/Enable MEM.SAV

The "N" command takes a binary load file from the disk and loads it into memory, just as does the "L" command. The difference is this command enables the MEM.SAV file before loading (and executing) the program; and when control is returned to MYDOS, the contents of memory will be saved back to MEM.SAV.

To just enable MEM.SAV, first enter this DOS menu (by typing "DOS" in BASIC), then press "N" and [RETURN].

O. System and Drive Configuration

The "O" command is used to specify the type (at least logically) of the disk drives on your computer and is also used to specify several system configuration parameters such as the number of file buffers provided or whether a RAMdisk is present. The system configuration functions, which are not specific to individual drives, are selected by pressing [RETURN] at the prompt "Drive number or RETURN:". The following table explains all the prompts for system configuration:

Verify WRITEs?	N=do not write with verify, else do verify
Number of File Buffers?	The number of disk files that can be open at once (RETURN = 3)
RAM disk present?	N=no RAM disk on the system (skips the next 4 questions as well), else the RAMdisk is enabled, and the next questions determine what kind it is.
[A]xlon or [X]E type?	A=Axlon, (RETURN)=XL or XE type. If Axlon, skip the next question.
Use default config for (x)K?	N=no, specify parameters. "Y" or [RETURN] uses the parameters the DOS generated for your type of machine. Note that the "x" is replaced by the amount

	of total memory MYDOS thinks you have. Skip the next three questions if answered yes.
Size(K)?	Size of the RAMdisk area in K (8 single density sectors = 1K); this must be a multiple of 16 (NI 256K upgrade=192 OR 128, AXLON=112, default or 130XE=64).
Control Address(HEX)?	Address of the memory map control port: a RETURN specifies $D301 for the 130XE. For the AXLON RAMPOWER 128, enter "CFFF" here.
Page sequence?	[RETURN] if using a standard Atari 130XE or equivalent RAMdisk. Other standard values are (1) for the Newell industries upgrade, (2) for the RAMBO from ICD, and (5) for Axlon or compatible boards. The most general response is to list the page numbers to be used as 2 hex digits each, separated by commas, and continuing if necessary by ending the line with a comma. The sequence is ended with a "0" if the RAMdisk is 130XE compatible, and with a "FF" if it is Axlon compatible.
RAM disk drive no?	Drive number used to access the RAMdisk (RETURN for drive 9 or a drive number 1-9)

A [RETURN] does not retain the current value when responding to these questions; it changes the setting to a generally acceptable default. Even a BREAK or RESET will not leave the configuration unchanged. If the configuration process is aborted for any reason, and you wish to retain your old configuration, you should reboot the computer or properly complete the configuration before doing anything else.

If instead of a [RETURN], a drive number had been specified, then that drive would be reconfigured. The first question identifies whether the drive is to be included in system initialization (and thus be available for later use). If a non-existent disk is included, it does not cause any problems with the system: it simply causes that disk to be examined each time the system is booted and reset (adding perhaps a second to the time it takes to boot MYDOS). If drive is excluded from the system, no further questions are asked. Otherwise, the second question asks if the drive is configurable: that is, is it like the ATARI 810 drive (with a fixed configuration) or is it like the PERCOM or ATR8000 drives. If the disk is not configurable it is assumed to be a 720 sector, single or double density ATARI 810/815/1050 disk drive.

The third question is whether the drive is a high capacity drive (does it use the modified configuration for non-floppy disks). This is for hard disks. This should be answered "N" if the drive is a

floppy disk drive, and "Y" if otherwise. If this question is answered "Y", the only remaining question is the size (in sectors, from 256 to 65535) of the drive. Drives up to 16 million bytes (Megabytes) capacity can used with MYDOS, and larger drives can usually be partitioned to appear as two or more drives of less than 16 million bytes capacity. See your hard disk manual for more information on this subject.

Drives that have not been configured can still be accessed. This configuration is necessary ONLY when formatting hard disks, and when using PERCOM or ATR8000 interfaces. Other than the above, the drive configuration is not necessary - MYDOS will figure out by itself what the disk is capable of by its format.

If the first three answers are "N" (do not exclude the drive), "Y" (it is configurable), and "N" (it is not a high capacity drive), the configuration is asked for: Is the drive double sided, how many tracks are there on the each side of the disk, and at what speed can it move the read/write head across the disk (what is its step rate). The first question is answered with "Y" or "N" ("Y" meaning "yes" it is a double sided drive, and diskettes formatted on it will be double sided). The second question is answered with 35, 40, 77, or 80 followed by a [RETURN]. If you have IBM PC/AT style high capacity disk drives (1.2Mb capacity) you may use 77 of the 80 tracks in high capacity mode providing 1 Mb capacity by selecting 77 tracks. No other numbers are accepted. The answer to this question specifies both the type of drive (8 inch/high capacity 5 1/4 inch, or standard 5 1/4 inch) as well as the number of tracks per side. This answer is very important to the operation of the drive.

The last answer ("Step Rate?") is entered as a code: use the following table and the drive specifications to determine the proper value:

Code value	8 inch rate	5 1/4 inch rate	
0	3 ms/track	6 ms/track	(XF-551)
1	6 ms/track	12 ms/track	
2	10 ms/track	20 ms/track	
3	15 ms/track	30 ms/track	

P. Diskette Density Selection

The density used for most MYDOS commands is determined by the data written on the diskette and the operator need not worry about setting it. The "P" command is provided to allow forcing the density setting for the format ("I") command, and to permit setting the density for disk drives and systems that do not automatically recognize the density of a diskette when reading sector 1 (for example, Indus GT drives).

Type the drive number, followed by "S" or "D" to set the density of the drive. Example: "2,D".

Q. Create Subdirectories

When a diskette is formatted, an empty directory (the highest level or root directory) is created. This directory is capable of holding up to 64 files or other directories. If additional directories are installed in this directory, each of the additional directories can contain up to 64 files as well. A subdirectory is installed in an existing directory using the "Q" command and responding with the filename of the new directory. For example, if "TEST" and "BAS" are two subdirectories in the root directory of the diskette in drive 1, "1:TEST:COMM" or "1:BAS>COMM" would create a new directory in the "TEST" or "BAS" directories respectively. "1:NEW:COMM" would not create a directory, however, since the directory "NEW" does not already exist. A "Q" command with the response "1:NEW" would create it, however, and a second "Q" command with the response "1:NEW:COMM" would then create the new directory called "COMM" within the directory "NEW".

Each directory takes up 8 sectors, and after it is created it may only be referenced as a directory (followed by a ":" or ">"), or deleted. It may only be deleted if it is empty (if it has no files in it). A directory may be emptied by using the "D" (delete) command and specifying the files "*.*" or "*.*/N" to remove all the files in the directory. You can most safely do this by specifying the full name of the directory: D5:MSGBASE>SEPT1983>*.*/N for example. Now to delete the SEPT1983 subdirectory, simply delete "D5:MSGBASE>SEPT1983" (note no trailing ":" or ">").

R. Set the Default Directory

The "R" command is used to select a directory to be used when a file is referenced without the drive number: that is, when file names such as "TEST1.BAS" or "D:NEWCODE" or even ":BIGFILE" are used, they are assumed to be in the default directory. Programs run under MYDOS can access the contents of the current default directory by using a file name of the form "D:..." without the drive number explicitly entered. They may also set the default directory by calling the CIO Function code 41 (set directory) routine.

The directory is set by inserting the diskette with that directory on it into the desired drive, then entering the file name of the directory with no trailing ":" or ">".

If the diskette in the drive containing the default directory is replaced, or if the default directory on a diskette is deleted, the default should be redefined. The one exception is if the default directory is the root or main directory on a diskette: this is because the root directory is at the same location on all diskettes.

V. FILE MANAGER FUNCTIONS PROVIDED THROUGH CIO

This part of the documentation is included in the technical notes of MYDOS, so consult those notes for details.

VI. CIO FUNCTION CODES PROVIDED BY MYDOS 4.50

This part of the documentation is included in the technical notes of MYDOS, so consult those notes for details.

VIII. MYDOS MEMORY MAP

This part of the documentation is included in the technical notes of MYDOS, so consult those notes for details.

IX. CUSTOMIZING A SYSTEM DISK

Much of this section is covered in the technical notes, but here are some things pertinent to all users.

RAMdisk Configurations

The RAMdisk driver included in MYDOS 4.50 will automatically configure itself on XE compatible memory upgrades. It is suggested you use this feature. If you REALLY want to set up the system to your specifications (such as configuring it to NOT use the banks BASIC XE uses), the power is available. This is necessary when AXLON compatible upgrades for the 800. The default page sequence uses the standard XE banks first.

The file RAMBOOT.M65, the MAC/65 assembler source code for the file RAMBOOT.AUT, is an AUTORUN.SYS file that simulates the operation of Atari DOS 2.5 and its RAMdisk handler. It "formats" the RAMdisk and copies DUP.SYS to it, as well as setting the RAMdisk unit number and the unit used to access the DUP.SYS and MEM.SAV files to 8.

By modifying the code in the source file and creating a modified AUTORUN.SYS file, the drive used to save MEM.SAV and fetch DUP.SYS can be modified, other files than just DUP.SYS can be copied to the RAMdisk when the system is booted, or any other operation could be performed that you find useful.

Number of Files Open at Once

The number of files that may be simultaneously open is set with the same byte as in ATARI DOS 2: location $0709 (decimal 1801). This byte contains a number from 1 to 16 setting the number of disk files that may be open at the same time. Normally it is set to 3, the smallest number that supports all the functions in the MYDOS menu. Specifically, a copy from one disk file to another requires three

open disk files. The value in the distributed version of MYDOS 4.50 is three; to permit more or fewer files, use the "O" command followed by a RETURN. To permanently change the maximum number of files, use the "H" command to write a modified MYDOS system to a disk. Each file that may be open at one time requires the allocation of a 256 byte buffer, so setting this value to 7 (instead of 3) will cause MYDOS to be 1024 bytes longer than before, and the programs loaded must begin no lower than $22E9 (instead of $1EE9). In corresponding fashion, by setting the value to 1, a BBS program can be loaded in with 512 bytes of additional memory if only one disk file is ever open at one time (commonly true of bulletin board programs).

Controlling the Disk Drives Accessed by MYDOS

Like ATARI DOS 2, MYDOS automatically identifies the disk drives that are present when booted up initially and any time it is reinitialized (some programs do this on exiting to the DOS and it is always done if the RESET key is pressed). MYDOS 4.50 is distributed with drives 1 and 2 configured, all others are omitted in order to speed up the booting process. To modify the maximum configuration MYDOS will use, invoke the "O" command for each drive to be added to (or removed from) the system. Pressing the RESET key will then use this value to redefine the drive. To permanently change the maximum drive configuration, use the "H" command to write out a new copy of MYDOS back to your system disk.

Selecting or Disabling Write-with-Verify

MYDOS 4.50 is distributed with the verify disabled, to speed up writing. Most drives (and disks) are very reliable, and function with no problems with the verify disabled. However, if you experience problems, or want to be 100% sure your data written will be good, use the "V" command to enable verify. As with DOS 2.0, the byte at location $0779 (1913 decimal) controls all write operations to the disk. If the value "poked" into it is $57 (87 decimal), then all writes will be read back to verify the action was successful. If the value "poked" into $0779 is $50 (80 decimal) then writes will be assumed successful, and will be performed in about one third the time taken when verify is enabled. Note that this address is not the same as in MYDOS 4.0 and 4.1 (where it was $0770 or 1904 decimal). To permanently alter the DOS, rewrite it back to the disk using the "H" command after changing it.

X. DISK DRIVE INTERFACE (via SIO)

This information is covered in the technical notes for MYDOS.

XI. RAMDISK INTERFACE

This information is also covered in the technical notes.

XII. INITIAL INSTALLATION INSTRUCTIONS

The following checklist is a procedure to bring up MYDOS on a new hardware configuration. It should be done with an un-modified copy of MYDOS 4.50, and care should be taken to perform the steps in exactly the order specified.

HOW TO GET FROM THE DRIVEWAY TO THE FREEWAY:

1. Before turning on your computer and disk drive, make sure that you do not have a cartridge installed in the computer.
2. Turn on the disk drive, and when it stops spinning, insert the MYDOS master diskette. Close the door.
3. Turn on the computer (with OPTION held if using an XL or XE computer). After several seconds, the MYDOS menu should appear on the screen. A prompt "Select Item (RETURN for menu):" will be displayed.
4. If you have a standard 2-drive system with single sided 40 track drives and either no RAMdisk or the standard Atari 130XE memory configuration, continue with step "6".
5. Configure each of your disk drives only if they are hard disks or special double-sided (XF-551) or more than 40 tracks by entering the "O" command followed by the drive number, and answering the questions the program asks you.
6. If you need to run a program that requires more than 3 disk files active at a time or you otherwise want alter one of the default parameters, press the "O" key followed by the RETURN key and answer the system configuration questions. This is also where the RAMdisk configuration needs to be entered if it is not a standard 130XE RAMdisk. ****
7. Now, insert a blank diskette in the disk drive and format it with the "I" command (remove the write protect tab if the diskette is write protected -- but first ask yourself why you put it there in the first place!). If an error message is displayed, insert another diskette; the first is bad.
8. Write the system files DOS.SYS and DUP.SYS to the newly formatted diskette with the "H" command and skip to step "10" if you do not have an ATARI 130XE computer or some other RAMdisk usable with the MYDOS RAMdisk driver.
9. If you have an ATARI 130XE computer or have another kind of RAMdisk and wish to use the MYDOS RAMdisk driver, copy RAMBOOT3.AUT on the master diskette to AUTORUN.SYS on the new diskette (use the "C" command, and if you have only one drive, enter the command line "RAMBOOT.AUT,AUTORUN.SYS/X" when MYDOS asks for the file names).**
10. This is now your primary system backup disk: place a write protect tab on it, label it, and use the "J" command to make as many working boot disks as you need from it.

NOTES:

The RAMBOOT3 program will first format your RAMdisk, then copy the files in a directory called RAMDISK on your boot disk to the RAMdisk if such a directory exists. To take advantage of this feature, use the "Q" command to create the directory and copy the files you want on the RAMdisk to it. If you do not want your ramdisk formatted each time you boot-up, which is not recommended, skip this step (step 9). Just be sure you format your ramdisk before using it.

** -- The source file for the RAM disk boot is also included; if you feel the urge, modify it to copy other files to the RAMdisk in addition to DUP.SYS (take out the MEM.SAV setting code, too, if you need the space) or perhaps add a call to change the default disk drive to D8:. Just remember that if you string together several autorun files, only the last will have its RUN address invoked.

**** -- NEWELL INDUSTRIES 800XL UPGRADE -- this RAMdisk can use page sequence 0, page sequence 1, or the default. The E3/E7/EB/EF pages can be reserved for BASIC/XE in extended mode by setting the RAMdisk size to 128K and the sequence to 1. Enter Size=128 (or 192 if you will not be using BASIC/XE often), Control address=D301, and Page sequence=1. In the second case this will use the area also used by BASIC/XE, but it will access it only after the first 128K are used up. By setting the size to 128K, BASIC/XE and the RAMdisk are fully separate and cannot interfere with each other.

**** -- AXLON RAMPOWER 128 for the Atari 800 computer -- MYDOS 4.50 comes configured for the Atari 130XE -- if you are installing it on an 800 with an Axlon RAMPOWER board, you must disable the 128K mode on the RAMPOWER board until you have configured the MYDOS RAMdisk code by entering Size=112, Control address=CFFF and Page sequence=5. Then write the DOS back to a new diskette, copy AUTORUN.SYS to it, switch the RAMPOWER board back to 128K and boot up the new diskette.

XIII. ERROR CODES AND THEIR SOURCES

3	Last byte of file read, next read will return EOF (MYDOS)
128	Break Abort (OS ROMs)
129	IOCB already open (OS ROMs)
130	No such device defined in the system (OS ROMs)
131	Write-only IOCB, cannot read (OS ROMs)
132	Invalid command (OS ROMs)
133	Device or File not open (OS ROMs)
134	Invalid IOCB reference (OS ROMs)
135	Read-only IOCB, cannot write (OS ROMs)
136	Attempt to read past end of file (MYDOS)
137	Truncated record (OS ROMs)
138	Device Timeout (OS ROMs)
139	Device NAK (serial bus failure, OS ROMs)
141	Cursor out of range for graphics mode (OS ROMs)
142	Data frame overrun (serial bus failure, OS ROMs)
143	Data frame checksum error (serial bus failure, OS ROMs)
144	Device I/O error (in peripheral hardware, OS ROMs)
146	Function not provided by handler (OS ROMs)
147	Insufficient RAM for graphics mode selected (OS ROMs)
160	Invalid Unit/Drive Number, zero or greater than 7 (both MYDOS and OS ROMs)
161	No sector buffer available, too many open files (MYDOS)
162	Disk full, cannot allocate space for output file (MYDOS)
163*	Write protected or system error - disk is not readable (MYDOS)
164	File number in link does not match the file's directory location (MYDOS)
165	Invalid file name (MYDOS)
166	Byte not within file, invalid POINT request (MYDOS)
167	File locked, cannot be altered (MYDOS)
168	Invalid IOCB (MYDOS and OS ROMs)
169	Directory full, cannot create a 65-th entry in a directory -- entries may be used for "lost" as well as real files (MYDOS)
170	File not in directory, cannot be opened for input (MYDOS)
171	IOCB not open (MYDOS and OS ROMs)
172*	File or directory of same name already exists in parent directory, cannot create (MYDOS)
173	Bad diskette or drive, cannot format diskette (MYDOS)
174*	Directory not in parent directory (MYDOS)
175*	Directory not empty, cannot delete (MYDOS)
180*	Not a binary file (MYDOS)
181*	Invalid address range for binary file, END<BEGIN (MYDOS)

* -- New error codes, not present or different in Atari DOS 2.

Most error codes are identical to those returned from ATARI DOS 2; the differences result from the expanded capabilities of MYDOS. Specifically, Error 164, indicating a file number mis-match, only occurs if the file is written in DOS 2.0 or DOS 2.5 format. Errors 180 and 181 can only occur when XIO 39 (or 40) is invoked to load a file. Errors 172 and 175 apply to creating and deleting directories

and have no ATARI DOS 2 equivalent; error 174 applies to accessing files in subdirectories, so it also has no ATARI DOS 2 equivalent. Error code 173 serves the same function as it did in ATARI DOS 2, but is returned more often (to identify bad diskettes more reliably).